Decision Support Systems 78 (2015) 15–27

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Multi-criteria analysis for OS-EMR software selection problem: A comparative study

A.A. Zaidan [a,b,*], B.B. Zaidan [b], Muzammil Hussain [b], Ahmed Haiqi [b], M.L. Mat Kiah [b], Mohamed Abdulnabi [b]

[a] Department of Computing, Faculty of Arts, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, Tanjong Malim, Perak, Malaysia
[b] Security Lab, Wisma R&D, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 15 June 2014
Received in revised form 18 June 2015
Accepted 3 July 2015
Available online 19 July 2015

Keywords:
Multi-criteria analysis
Open-source electronic medical record software
Multiple-criteria decision-making

ABSTRACT

Various software packages offer a large number of customizable features to meet the specific needs of organizations. Improper selection of a software package may result in incorrect strategic decisions and subsequent economic loss of organizations. This paper presents a comparative study that aims to evaluate and select open-source electronic medical record (OS-EMR) software based on multiple-criteria decision-making (MCDM) techniques. A hands-on study is performed, and a set of OS-EMR software are implemented locally in separate virtual machines to closely examine the systems. Several measures as evaluation bases are specified, and systems are selected based on a set of metric outcomes by using AHP integrated with different MCDM techniques, namely, WPM, WSM, SAW, HAW, and TOPSIS. Paired sample t-test is then utilized to measure the correlations among different techniques on ranking scores and orders. Findings are as follows. (1) Significant differences exist among MCDM techniques on the basis of different integrations on ranking scores, whereas no significant differences exist among them when representing the ranking scores to the ranking orders in place of the technique scale. (2) The software GNUmed, OpenEMR, OpenMRS, and ZEPRS do not differ in ranking scores/orders of experiments for all MCDM techniques presented. On the contrary, discrepancies among the ranking scores/orders are more noticeable in other software. (3) GNUmed, OpenEMR, and OpenMRS software are the most promising candidates for providing a good basis on ranking scores/orders, whereas ZEPRS is not recommended because it records the worst ranking score/order in comparison with other OS-EMR software.

1. Introduction

The vast diversity among available open-source software makes it difficult for customers to decide on which software to use [1,2]. Such decision problems have attracted considerable attention from the industry and academia. For example, Loyola and Franklin explicitly applied MCDM to assist with decision making. MCDM is modeled after the way humans are thought to make decisions. Although different MCDM methods, techniques, and approaches have been studied, the basic components of MCDM are the same: a finite or infinite set of actions, at least two criteria, and one decision maker. With these elements, MCDM assists in decision making mainly by choosing, ranking, or sorting the actions [3]. Consequently, MCDM is not only a collection of theories, methodologies, and techniques but also a specific perspective in dealing with decision-making problems [4]. Over the past few decades, decision-making theory has been successfully applied to a growing number of diverse domains and has assisted in decision making. MCDM can handle multiple conflicting criteria [5].

The applications explored in this study, for which many open-source options are already available, are deemed to have the potential to play a key role in applying informatics to healthcare and serve as alternatives to existing and deployed commercial/proprietary systems. Different types of applications exist, and they are as numerous as existing healthcare sub-fields and activities. The project categories listed in the online open-source healthcare repository, Medical Free/Libre and Open-source Software, support this concept [17,18]. An example of application classification in the literature is given in [19], in which the authors defined 15 functional classes specific to the medical domain based on the predominant functionality of the projects. These classes include the following: clinical information system/electronic medical record (EMR) system, messaging, continuing medical education, data acquisition, decision support, imaging, issue tracking, laboratory information system, and telemedicine. In this study, we consider only the first application type, that is, open-source electronic health record (EHR)/EMR software packages. This application is also the most common application used. According to the International Organization of Standards, EHR is defined by ISO/DTR 20514 as "a repository of information regarding the health of a subject of care in computer process able form, stored and transmitted securely, and accessible by multiple authorized users" [20]. EMR is often used in parallel with EHR, sometimes interchangeably,

* Corresponding author.
E-mail address: aws.alaa@gmail.com (A.A. Zaidan).

http://dx.doi.org/10.1016/j.dss.2015.07.002
0167-9236/

but many times within a confined context of a patient in a single healthcare organization or even a single encounter or care episode. According to this view, EMR is a point-in-time view of a larger EHR [21].

This study presents a new methodology to help decision-making processes in OS-EMR software selection. A comparative study is conducted to evaluate and select OS-EMR software based on MCDM techniques. The remaining sections of this paper are organized as follows. Section 2 covers the literature review. Section 3 describes the decision-making methodology for the evaluation and selection of OS-EMR software. Section 4 presents the comparison results and discussion. Sections 5 and 6 discuss the limitations and contributions of this research, respectively. Section 7 concludes the paper.

2. Literature review

The currently available OS-EMR software application have not been adequately analyzed and compared with guide potential implementers. Commonly used decision making strategies can be used to compare available OS-EMR software application. In any multiple-attribute decision making (MADM) ranking, fundamental terms need to be defined, including the decision matrix (DM) or the evaluation matrix (EM), the alternatives, and the criteria [6]. The evaluation matrix that consists of m alternatives and n criteria needs to be created; with the intersection of each alternative and criteria given as x_{ij}, we have a matrix $(x_{ij})_{m*n}$.

$$DM/EM = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \dots & C_n \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \end{array}$$

where $A_1, A_2, \ldots, A_m$ are possible alternatives that decision makers have to score (i.e., OS-EMR software); $C_1, C_2, \ldots, C_n$ are criteria against which each alternative's performance is measured (i.e., installation process, security, user support, developer support, customizability, functionality, and usability). Finally, x_{ij} is the rating of alternative A_i with respect to criterion C_j, W_j is the weight of criterion C_j (i.e., customizability weight, functionality weight, and usability weight). Certain processes need to be completed to rank the alternatives, such as normalization, maximization indicator, adding weights, and other processes depending on the method. Based on the literature [7,8], this review features the popular methods for MCDM, highlights their benefits and drawbacks, and provides recommendations, as shown in Fig. 1.

2.1. Analytic hierarchy process (AHP)

AHP is a multi-criteria decision making approach for dealing with complex decision problems. It is a multilevel structured technique that provides a comprehensive framework for evaluating different alternative solutions to a certain problem. By defining objectives, criteria, sub-criteria, and alternatives of a decision problem, AHP provides alternative solutions. It first decomposes the decision problem into different criteria. If a criterion is complex, then AHP further decomposes it into sub-criteria, and so on. Afterwards, each criterion is analyzed independently. Once the hierarchy has been constructed, AHP analytically evaluates the different criteria by comparing them with one another. AHP uses a pairwise comparison technique to evaluate different alternatives. Pairwise comparisons define the relative importance of each alternative in reference to each criterion. From the pairwise comparison, AHP extracts weights of importance of each criterion. On the basis of each criterion, AHP measures the performance of each alternative. AHP then transforms these assessments into numerical values and then uses these numerical values to determine the priorities of each alternative. The final decision is made on the basis of these priorities. To compare different attributes, we need a scale that describes how many times more or less important one element is from another [31].

2.2. Weighted sum model (WSM)

In operation research, WSM is a well-known and the simplest MCDM method for evaluating a number of alternatives in terms of a number of criteria. In general, suppose that a given MCDM problem is defined on m alternatives and n decision criteria, all the criteria are benefit criteria (i.e., higher values correspond to a better result), w_j denotes the relative weight of importance of the criterion C_j, and a_{ij} is the performance value of alternative A_i when it is evaluated in terms of criterion C_j; then, the total (i.e., when all the criteria are considered simultaneously) importance of alternative A_i, is defined as follows:

$$A_i = \sum_{j=0}^{n} W_i a_{ij} \quad \text{For}\, i = \{1, 2, \ldots ..m\}. \tag{1}$$

For the maximization case, the best alternative is the one that obtains the maximum total performance value.

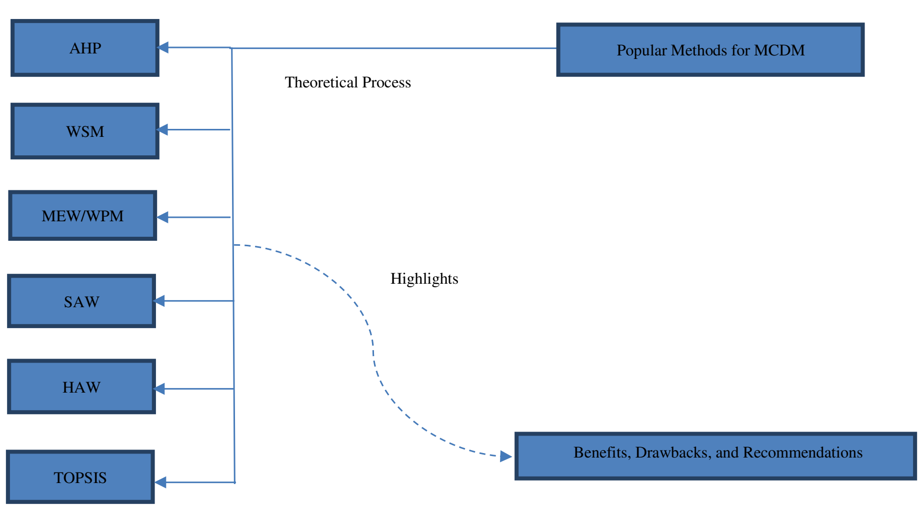

Fig. 1. Literature review framework.

2.3. Multiplicative exponential weighting (MEW) or weighted product method (WPM)

WPM is one of the simplest and earliest MCDM techniques. It is almost similar to WSM; the only difference between both methods is that addition is the main mathematical operation in WSM, whereas multiplication is the main mathematical operation in WPM. As with all MCDM methods, given a finite set of decision alternatives; these alternatives are measured against a number of criteria. Each alternative is scored by comparing the available alternatives by multiplying a number of ratios; each ratio is raised to the power equivalent to the relative weight of the corresponding criterion. Suppose that the MCDM problem is defined on m alternatives and n decision criteria, all the criteria are benefit criteria (i.e., higher values correspond to a better result), w_j denotes the relative weight of importance of the criterion C_j, and a_{ij} is the performance value of alternative A_i when it is evaluated in terms of criterion C_j; then, to compare the two alternatives A_K and A_L (where $m \geq K, L \geq 1$), the following product has to be calculated as follows:

$$P\left({}^{A_k}\!\Big/_{A_l}\right) = \prod_{j=1}^{n} \left({}^{a_{K_j}}\!\Big/_{a_{L_j}}\right)^{w_j}, \ \text{for} K,\ L = \{1, 2, \ldots ..m\}. \tag{2}$$

If the ratio $P(A_K/A_L)$ is greater than or equal to the value 1, then alternative A_K is more desirable than alternative A_L (in the maximization case). Therefore, the best alternative is that which is better than or at least equal to all other alternatives.

2.4. Simple additive weighting (SAW)

SAW, which is also known as the weighted sum method, is probably the best known and most widely used MADM method. The basic logic of SAW is to obtain the weighted sum of the performance ratings of each alternative over all attributes by performing the following steps:

Step 1: *Linear scale transformation*

In this step, the value of the criterion is divided by the maximum value of the criterion for all alternatives. Therefore, the following alternatives are obtained:

$$r_{ij} = \boldsymbol{x_{ij}}/x_j^*, \tag{3}$$

$$r_{ij} = \ \min x_j/\boldsymbol{x_{ij}}, \tag{4}$$

Eq. (3) is used when the criteria are benefit criteria, whereas Eq. (4) is used when the criteria are cost criteria. This process will result in the new matrix R.

$$R = \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ r_{21} & r_{22} & \ldots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \ldots & r_{mn} \end{bmatrix}.$$

Step 2: *Construction of the weighted transformed decision matrix*

In this step, a set of weights, i.e., $w = w_1, w_2, w_3, \cdots, w_j, \cdots, w_n$, from the decision maker is accommodated in the transformed decision matrix. The resulting matrix can be calculated by multiplying each column in the normalized decision matrix (R) with its associated weight w_j. As mentioned before, the set of the weights is equal to 1. This process will result in a new matrix V as shown below.

$$V = \begin{bmatrix} v_{11} & v_{12} & \ldots & v_{1n} \\ v_{21} & v_{22} & \ldots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & \ldots & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \ldots & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \ldots & w_n r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_1 r_{m1} & w_2 r_{m2} & \ldots & w_n r_{mn} \end{bmatrix}.$$

Step 3: *Construction of the weighted average value for the alternatives*

In this step, the summation of the new values that resulted from the previous step is calculated as

$$A_i^* = \sum_{j=1}^{6} v_{ij}\ ,\ j = \{1, 2, \ldots ..n\}. \tag{5}$$

Step 4: *Ranking the alternative according to descending values of the alternatives*

The set of the alternative A can now be ranked according to the descending order of the alternatives, where the highest value indicates the best performance.

2.5. Hierarchical adaptive weighting (HAW)

In SAW, each criterion value is divided by the largest criterion value among all alternatives. Unlike SAW, the HAW method is composed of the following steps:

Step 1: *Rescoring*

In this step x_{ij} is used as the ratio of as the sub-score of the alternative ith with regard to the jth criterion, which is defined as

$$k_j = x_{ij} \Big/ \sum_{i=1}^{4} x_{ij}\ , j = \{1, 2, \ldots ..n\} \tag{6}$$

$$k_j = \frac{1}{x_{ij}} \Big/ \sum_{i=1}^{4} \frac{1}{x_{ij}}\ ,\ j = \{1, 2, \ldots ..n\}. \tag{7}$$

Eq. (6) is used when the criteria are benefit criteria, whereas Eq. (7) is used when the criteria are cost criteria. This step results in the new matrix K.

$$K = \begin{bmatrix} k_{11} & k_{12} & \ldots & k_{1n} \\ k_{21} & k_{22} & \ldots & k_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ k_{m1} & k_{m2} & \ldots & k_{mn} \end{bmatrix}.$$

Step 2: *Ranking the alternatives based on the mission effectiveness*

Assume that the set of weights, i.e., $w = w_1, w_2, w_3, \cdots, w_j, \cdots, w_n$, from the decision maker is accommodated to compute the vector for the hierarchical mission effectiveness h is given by

$$h = k * w^T \tag{8}$$

where (w^T) is the transpose of vector (w).

Step 3: *Ranking the alternative according to the descending values of the alternatives*
The set of the alternative A_i can now be ranked according to the descending order of the alternatives, where the highest value indicates the best performance.

2.6. Technique for order performance by similarity to ideal solution (TOPSIS)

TOPSIS allocates scores to each alternative on the basis of their geometric distance from the positive and the negative ideal solutions. According to this technique, the best alternative would be the one with the shortest geometric distance to the positive ideal solution and the longest geometric distance to the negative ideal solution determined by going through the following steps:

Step 1: *Construction of the normalized decision matrix*
In this step, various dimensional attributes are transformed into non-dimensional attributes to allow comparison of the attributes. The matrix $(x_{ij})_{m\ *\ n}$ is then normalized from $(x_{ij})_{m\ *\ n}$ to the matrix $R = (x_{ij})_{m\ *\ n}$ by using the normalization method.

$$r_{ij} = x_{ij} \Big/ \sqrt{\sum_{i=1}^{m} x_{ij}^2} \quad \text{for all} \quad i = 1\ldots..n \ \text{ and } \ j = 1\ldots\ldots..n^m. \tag{9}$$

This process will result in a new matrix R, which is shown below.

$$R = \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ r_{21} & r_{22} & \ldots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \ldots & r_{mn} \end{bmatrix}.$$

Step 2: *Construction of the weighted normalized decision matrix*
In this process, a set of weights, i.e., $w = w_1, w_2, w_3, \cdots, w_j, \cdots, w_n$ where $j = 1,\ldots,n$ from the decision maker is accommodated in the normalized decision matrix. The resulting matrix can be calculated by multiplying each column from the normalized decision matrix R with its associated weight $_{wj}$. The set of the weights is equal to 1.

$$\sum_{j=1}^{m} w_j = 1. \tag{10}$$

This process will result in a new matrix V, which is shown below.

$$V = \begin{bmatrix} v_{11} & v_{12} & \ldots & v_{1n} \\ v_{21} & v_{22} & \ldots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & \ldots & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \ldots & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \ldots & w_n r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_1 r_{m1} & w_2 r_{m2} & \ldots & w_n r_{mn} \end{bmatrix}.$$

Step 3: *Determining the ideal and negative ideal solutions*
In this step, two artificial alternatives A^* (the ideal alternative) and A^- (the negative ideal alternative) are defined as

$$A^* = \left\{ \left(\left(\max_i v_{ij} \middle| j \in J \right), \left(\min_i v_{ij} \middle| j \in J^- \right) \middle| i = 1, 2, \ldots\ldots\ldots\ldots, m \right) \right\} = \left\{ v_1^*, v_2^*, \ldots, v_j^*, \cdots v_n^* \right\} \tag{11}$$

$$A^- = \left\{ \left(\left(\min_i v_{ij} \middle| j \in J \right), \left(\max_i v_{ij} \middle| j \in J^- \right) \middle| i = 1, 2, \ldots\ldots\ldots\ldots, m \right) \right\} = \left\{ v_1^-, v_2^-, \ldots, v_j^-, \cdots v_n^- \right\} \tag{12}$$

where J is a sub-set of $\{i = 1, 2, \ldots m\}$, which presents the benefit attribute (i.e. which offers an increasing utility with its higher values), whereas J^- is the complement set of J and the opposite could be added for cost type attribute as well, as denoted by J^c.

Step 4: *Separation measurement calculation based on the Euclidean distance*
In this step, the separation measurement is performed by calculating the distance between each alternative in V and the ideal vector A^* using the Euclidean distance, which is given by

$$S_{i^*} = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^* \right)^2}, \quad i = \{1, 2, \ldots..m\}. \tag{13}$$

Similarly, the separation measurement for each alternative in V from the negative ideal A^- is given by

$$S_{i^-} = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^- \right)^2}, \quad i = \{1, 2, \ldots..m\}. \tag{14}$$

By the end of step 4, two values, namely, S_{i^*} and S_{i^-}, for each alternative have been counted. These two values represent the distance between each alternative and both the ideal and the negative ideal.

Step 5: *Closeness to the ideal solution calculation*
In this step, the closeness of A_i to the ideal solution A^* is defined as

$$C_{i^*} = S_{i^-} / \left(S_{i^-} + S_{i^*} \right), \ 0 < C_i < 1, \quad i = \{1, 2, \ldots..m\}. \tag{15}$$

Obviously, $C_{i^*} = 1$ if and only if $A_i = A^*$. Similarly, $C_{i^*} = 0$ if and only if $A_i = A^-$

Step 6: *Ranking the alternative according to the closeness to the ideal solution*
The set of the alternative A_i can now be ranked according to the descending order of C_{i^*}; the highest value indicates the best performance.

According to the literature [7–10,20] the benefit and drawbacks of and recommendations for the above MCDM techniques can be summarized as follows: WSM and HAW techniques are easy to use and understand, but weights to the attributes are assigned arbitrarily, and both techniques become difficult to use as the number of criteria increases. Another problem with these methods stems from the use of common numerical scaling to obtain final score. SAW considers all the criteria, makes decisions intuitively, and offers simple calculation; however, all the values of the criteria should be maximum and positive. Moreover, SAW does not always reflect the real situation. With regard to MEW/WPM, its strong points include its ability to remove any unit of measure and the use of relative values rather than actual ones; however, no solution with equal weight of decision matrices is offered. AHP enables

decision makers to structure a decision making problem into a hierarchy, which helps them to understand and simplify the problem. However, it is a time-consuming technique because of the mathematical calculations and number of pairwise comparisons, which increase as the number of alternatives and criteria increases or changes. Ranking of the alternatives in AHP depends on alternatives considered for evaluation. Adding or deleting alternatives can change the final ranking (rank reversal problem). TOPSIS is functionally associated with problems of discrete alternatives. It is one of the most practical techniques for solving real-world problems. A relative advantage of TOPSIS is its ability to identify the best alternative quickly. The major weakness of TOPSIS is its lack of provision for weight elicitation and consistency checking for judgments. However, the use of AHP has been significantly restrained by the human capacity for information processing; thus, the 7 ± 2 would be the ceiling for comparison [21]. From this viewpoint, TOPSIS reduces the required paired comparisons, and the capacity limitation might not significantly dominate the process. Hence, TOPSIS is suitable for cases with a large number of attributes and alternatives, and it is especially handy when objective or quantitative data are given. In fuzzy-based approach, decision makers can use linguistic terms to evaluate alternatives that improve the decision making process by accommodating the vagueness and ambiguity in human decision making. However, computing fuzzy appropriateness index values and ranking values for all alternatives is difficult.

MCDM techniques discussed in this section help rank the candidate software packages and choose the best one, but they lack indicators of how well each candidate software package can meet user needs. Another problem with these techniques is its non-adoption of a requirement-driven approach, which makes them inadequate for software selection decision making [23]. The recommendation after reviewing the relevant literature is the integration of AHP with WPM/WSM/SAW/HAW/TOPSIS, each of which is widely adopted for selecting software packages [22,24,30].

3. Methodology

3.1. Conceptual framework

Our study provides a detailed look at the alternative systems based on a set of measures via the relatively infrequent route of actually installing the software and reporting more hands-on information on the installation process, usability, and other variables. The input to this part (sources and inclusion criteria of subject articles) is discussed in later sub-sections.

The selection of OS-EMR software received input from both the preliminary implementation of each system in a virtual machine and online information from the systems' websites. The output is the comparison between the subject systems based on our set of variables, using different integrations between AHP and the other MCDM/MADM techniques, i.e., WPM/WSM/SAW/HAW/TOPSIS, as well as more insights into the limitations and merits of each system. All the elements of our study are shown in the overall conceptual framework in Fig. 2.

3.1.1. Evaluation and selection of OS-EMR software

The evaluation and selection of the OS-EMR software involves procedures and steps that a decision maker follows during software selection decision making. This methodology is not intended to be a rigid structure; it only serves as a guide or aid that can be adapted according to the requirements of the individual organization. On the basis of the literature review, we have developed a generic stage-based methodology for the selection of OS-EMR software packages. This methodology comprises the following three phases:

- *Preliminary investigation of the availability of OS-EMR software*
  Several elements limit the scope of our study. The term "health informatics software" in this paper applies only to a sub-set of health informatics, a sub-set of software systems, and a sub-set of the combination of both sub-sets [12–16]. Health informatics comprises a group of fields, each with different software applications; in this paper, we consider only the category of electronic medical records and electronic health records [17–19]. This selection excludes other informatics fields, such as bioinformatics, and other health informatics sub-fields, such as imaging and visualization [11]. Furthermore, the sole focus of this paper is to analyze the available options among OS-EMR software exclusively, as opposed to proprietary or commercial systems.
  Finally, we selected 13 available OS-EMR software packages listed in the online open-source healthcare repository Medical Free/Libre and Open-source Software [17,18]. An example of application classification in the literature is given in [19]. Active OS EHR/EMR systems include FreeMED, GNUmed, GNU Health, Hospital OS, HOSxP, OpenEMR, OpenMRS, OSCAR, THIRRA, WorldVista, ZEPRS, ClearHealth, and MedinTux. This list does not include all the available OS EHR/EMR systems but is a

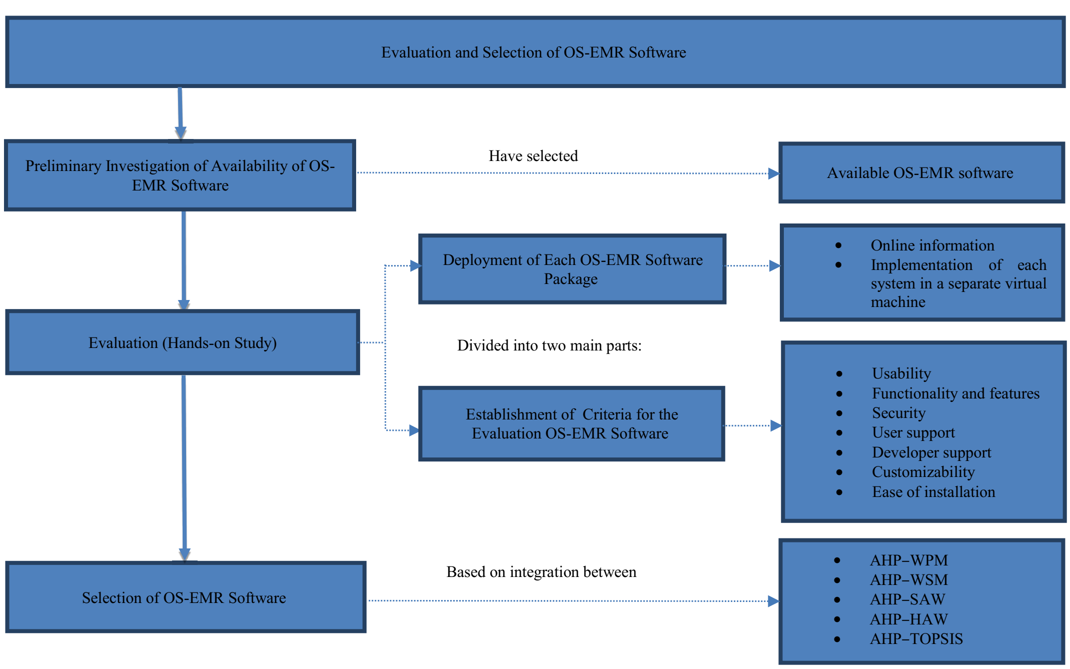

Fig. 2. Conceptual framework.

representative set of popular OS EHR/EMR systems in the literature as well as in open-source repositories.

- *Evaluation (hands-on study)*
  The hands-on study is divided into two main parts, namely, deployment of each OS-EMR software package and establishment of criteria for evaluation.

3.1.1.1. Deployment of each OS-EMR software package. This phase involves two primary sources of information. First, available information and details about the products, including their online documentation and resources, are found online, mostly from the systems' own websites. Second, the actual preliminary implementation of each system in a separate virtual machine allows us to experience the installation process and obtain first-hand impression about the usability of the system. Such implementation can serve as a thorough investigation of the systems that are useful for future research.

Our approach was to install each system on a separate virtual machine. For that purpose, we experimented with VMware Player and VirtualBox open virtualization packages. Then, delivered images of the installed software were produced using VirtualBox exclusively because the free VMware edition does not export applications. In each case, we initially created the virtual machine and installed the guest operating system. We did not consider the setting of the computer system in which to run the healthcare system as part of the installation process of that system. To serve as a convenient reference, we also captured the processes on video screen records.

This report summarizes our major findings about the systems based on a set of measures. The aim of this study is to provide a preliminary review of the comparative features of the systems, and help researchers and particularly potential implementers understand available open options in the healthcare information systems industry. The measures against which we recorded the comparisons are summarized in Table 1.

3.1.1.2. Establishment of criteria for the evaluation of OS-EMR software. In this phase, criteria to be used for the evaluation of OS-EMR software are identified and arranged. In review articles, the common metrics used to compare open-source software (OSS) systems include technical details such as platform, programming language, and database system [34,35], functions/features, and security [36,37]. To provide a broader scope of evaluation, we added a few other measures. Referring to the literature on software quality models, a frequently referenced modeling standard is the International Organization for Standardization and International Electro-technical Commission 9126-1 quality standard [38]. ISO/IEC 9126-1 specifically defines a quality model and its use as a framework for software evaluation. A 9126-1 quality model is defined by means of general software characteristics, which are further refined into sub-characteristics, which in turn are decomposed into attributes [39]. The six quality characteristics defined in the 9126-1 quality standard are functionality, reliability, usability, efficiency, maintainability, and portability.

Reliability and maintainability metrics are hard to measure in a limited experimental setting, and require long-term real-world evaluation.

Table 1
Reference measures against which OSSs were compared.

Measure	Description
Usability	How efficient, easy, and convenient is the system to learn and use?
Functionality and features	What core functions does the system include with respect to a common reference set?
Security	What security measures does the system support?
User support	How well does the system help its users?
Developer support	How well does the system allow for new developers to modify or contribute to it?
Customizability	How much can the system be altered to suit different users' needs or preferences?
Ease of installation	How easy can the system be installed before actual use?

Also, almost all our representative OSS are cross-platform, thereby excluding the need to include portability as a comparison factor. However, we kept the ease of installation metric (a measure under the sub-characteristic of portability quality in the 9126-1 standard) to emphasize the implementer perspective in our study. We also chose to highlight customizability from the usability measure and user support as separate factors in evaluating OSS programs [40].

The procedure of the OS-EMR software evaluation is presented in the following sub-sections. Each measure is established as a sub-set of elements. The comparisons results among software based on the seven measures are listed in Table 1. Scores of systems are based on those elements.

3.1.1.2.1. Usability. The following set of measures was adopted as basis for comparison among the usability of the systems:

- Efficiency (time to accomplish tasks or throughput)
- Learnability (time to learn and ease of learning)
- Satisfaction (comfort and acceptability of use).

These measures were selected from a number of usability attributes in various standards and models reported in [32]. Efficiency and satisfaction metrics were adopted from ISO 9241-11 (1998) standard [42], and learnability was adopted from several books and studies [43–45]. Some common functions were referred to and a rudimentary experiment was conducted in which specific tasks are attempted on the systems; a quantitative index was given per measure. The scale of the index was normalized to a range of 1 to 5. Three users were selected for this test; two are from the medical field with low and high IT literacy, and the third user is a professional programmer. In each test, the thirteen systems were presented to the subject test user in its respective virtual machine. Two of the researchers evaluated the usability, provided needed instructions, timed, and then questioned the test subjects. The following are three common tasks in an electronic medical records system: (1) create a patient, fill the basic demographic data, (2) enter a visit at the counter, and (3) generate a report. The test users were asked to determine how these tasks can be fulfilled without assistance other than online help and documentation. The time consumed during this process was measured. The time indicated learnability of a given system compared with another based on the same given tasks. After the user learned how to accomplish the tasks, the actual time taken to actually perform the operations was recorded to measure the relative efficiency associated with each application relative to the others. Finally, the subjective feeling of ease and convenience experienced by the user, as well as his impression of the user interface, gauged the satisfaction factor [27]. Table 2 reports the results of the comparison. We were aware of the preliminary nature of this test. Our intention was to expand the usability measure into a separate study with an adequate sample size. The goal was to provide a taste of the relative satisfaction of each system based on a small reference sample.

3.1.1.2.2. Functionality and features. Most of the considered systems shared a common set of core functionality but widely vary in the finer features. The more granular features were classified into four domains, namely, patient handling, medical records, support data, and admin tools. Each domain included sub-features numbered from 1 to 25, as shown in Table 2. These features were distributed on the open-source EMR systems as follows: OpenEMR (19/25), GNUHealth (16/25), GNUmed (14/25), WorldVista (14/25), OSCAR (14/25), FreeMED (13/25), HOSxP (10/25), HospitalOS (9/25), MedinTux (9/25), OpenMRS (8/25), ClearHealth (7/25), THIRRA (7/25), and ZEPRS (5/25).

3.1.1.2.3. Security. An overall evaluation of the software systems' security is beyond the scope of this paper. Measurement of the security on many aspects requires extensive analysis. From a defense-in-depth perspective, security at the application and data levels is applicable for individual software systems, whereas security of outer layers (host and network security) is deployment dependent. Data security implies

Table 2
Evaluation matrix.

Software	Usability			Functionality and features	Security						User support								Developer support						Customizability					Ease of installation
	Learnability	Efficiency	Satisfaction		Secure authentication and access control	Secure storage	Secure exchange	Logging mechanism	Backup mechanism	HIPPA compliance	Website	Forum	Wiki/blog	Video/webcast	User guide	Installation guide	Demo site	Commercial support	Group	Mailing list	Code comments	Developer guide	Code repository	Bug tracker	Localization	Interface control	Add-ons ability	Customs scripts	Custom reports	Installation time (hours)
FreeMED	4	3	2	13/25	Y	N	N	N	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	N	Y	N	Y	N	N	N	N	N	N	4
GNUmed	3	4	4	14/25	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	2
GNU Health	4	4	3	16/25	Y	N	N	N	N	N	Y	N	Y	N	Y	Y	N	Y	N	Y	N	N	Y	Y	Y	Y	Y	N	N	2
Hospital OS	3	3	2	9/25	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	1
HOSxP	1	1	1	10/25	Y	N	N	N	Y	N	Y	Y	Y	Y	N	N	N	Y	Y	N	N	N	N	N	N	Y	N	Y	Y	5
OpenEMR	4	4	3	19/25	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	N	N	1
OpenMRS	3	4	3	8/25	Y	N	N	Y	N	N	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	3
OSCAR	3	3	3	14/25	Y	N	N	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	N	N	Y	N	Y	3
THIRRA	4	4	3	7/25	Y	N	N	N	N	N	Y	N	Y	N	Y	Y	Y	N	N	x	Y	N	Y	N	Y	N	N	N	N	2
WorldVista	3	3	3	14/25	Y	N	N	N	N	Y	Y	Y	N	N	Y	Y	Y	N	N	N	N	Y	N	N	N	N	N	N	N	4
ZEPRS	3	3	2	5/25	N	N	N	N	N	N	Y	N	N	N	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	1
ClearHealth	3	3	3	7/25	Y	N	N	N	N	Y	N	N	N	N	N	N	N	Y	N	N	Y	N	Y	N	N	N	Y	N	Y	2
MedinTux	1	1	1	9/25	N	N	N	N	N	N	Y	Y	Y	Y	N	Y	N	N	Y	N	N	N	N	N	N	Y	N	N	N	7

secure storage through encryption and protection against loss through backup policies. Application security implies a multitude of measures to ensure that no vulnerabilities in the application can be exploited to compromise an asset. Thorough assurance of application security requires a vulnerability scan, code analysis, and penetration testing.

In the area of health information systems, rigorous security requirements, particularly on medical records privacy, are expected. Few standards began to outline those requirements, most notable of which are the American Health Insurance Portability and Accountability Act (HIPAA) privacy rules. Out of the 13 systems under consideration, only four comply with the HIPAA. Compliance with security and privacy regulations has more components than just the software through which records are supplied and retrieved; it involves comprehensive adherence to policies, practices, measures, and tools that cover the entire life cycle of health information, including storage and exchange.

Nevertheless, to provide an indication of the comparable security of our examined systems, we adopted a sub-set of security metrics against which the systems were compared; the results are reported in Table 2. Access control, HIPAA compliance, and security of communication and data exchange were adopted from [28,47] based on security and privacy

Table 3
Results of weight calculated for four different developers.

Criteria	Sub-criteria	Weight calculated from the survey			
		Developer 1	Developer 2	Developer 3	Developer 4
Usability	Learnability	0.102078	0.003362	0.017036	0.049233951
	Efficiency	0.051006	0.030275	0.049347	0.049233951
	Satisfaction	0.010772	0.030275	0.003903	0.049233951
Functionality and features	Functionality and features	0.1478944	0.27895011	0.04261939	0.13003756
Security	Secure authentication and access control	0.033865	0.037264	0.007614	0.024601446
	Secure storage	0.055686	0.031322	0.005437	0.015261721
	Secure exchange	0.027609	0.062862	0.010347	0.015261721
	Logging mechanism	0.027609	0.10703	0.020468	0.015261721
	Backup mechanism	0.029901	0.076822	0.020468	0.012418779
	HIPAA compliance	0.02223	0.050086	0.039545	0.015261721
User support	Website	0.017431	0.003827	0.015303	0.010846816
	Forum	0.004769	0.003419	0.015339	0.014614091
	Wiki/blog	0.004769	0.003419	0.017118	0.004751382
	Video/webcast	0.007482	0.003003	0.016784	0.007520622
	User guide	0.004769	0.010997	0.007483	0.034435506
	Installation guide	0.004769	0.020418	0.012627	0.022464576
	Demo site	0.004769	0.013398	0.031819	0.004751382
	Commercial support	0.005509	0.001118	0.037847	0.007023898
Developer support	Group	0.003772	0.004068	0.002346	0.00717414
	Mailing list	0.012549	0.007147	0.004242	0.00717414
	Code comments	0.018585	0.02065	0.008222	0.064579292
	Developer guide	0.024126	0.001602	0.00038	0.014266205
	Code repository	0.037324	0.02065	0.001053	0.006386034
	Bug tracker	0.023426	0.02065	0.001934	0.012710633
Customizability	Localization	0.038161	0.00317	0.01162	0.056207623
	Interface control	0.027107	0.001472	0.191107	0.138187693
	Add-on ability	0.027107	0.006234	0.029122	0.036041038
	Custom scripts	0.014556	0.002241	0.057681	0.01277737
	Custom reports	0.038161	0.004668	0.094049	0.034903984
Ease of installation	Installation time (hours)	0.172209	0.13960158	0.22714075	0.12737706

Table 4
Scores based on different integrated AHP–(WSM/SAW/HAW/WPM/TOPSIS) for W1, W2, W3, and W4.

Based on integrated AHP–TOPSIS

Software	Scores with different developer weighted											
	S1+	S1−	W1	S2+	S2−	W2	S3+	S3−	W3	S4+	S4−	W4
FreeMED	0.0796	0.0496	0.3839	0.0996	0.0759	0.4325	0.1015	0.0352	0.2575	0.074	0.0434	0.3697
GNUmed	0.0698	0.0613	0.4676	0.0939	0.0847	0.4742	0.0311	0.1031	0.7683	0.0306	0.081	0.7258
GNU Health	0.0724	0.0631	0.4657	0.1046	0.0777	0.4262	0.0676	0.084	0.5541	0.0435	0.0762	0.6366
Hospital OS	0.0851	0.0428	0.3346	0.1232	0.038	0.2357	0.0706	0.0853	0.5471	0.063	0.0607	0.4907
HOSxP	0.0899	0.0357	0.2842	0.1143	0.0566	0.3312	0.0563	0.0957	0.6296	0.0659	0.0589	0.472
OpenEMR	0.0656	0.0783	0.5441	0.0705	0.1229	0.6355	0.0598	0.094	0.6112	0.0288	0.0878	0.753
OpenMRS	0.0771	0.0581	0.4297	0.1127	0.0674	0.3742	0.0521	0.0981	0.6531	0.0429	0.0775	0.6437
OSCAR	0.0743	0.0537	0.4195	0.0942	0.0835	0.4699	0.09	0.0564	0.3852	0.0718	0.0458	0.3895
THIRRA	0.086	0.0438	0.3374	0.1298	0.0286	0.1806	0.1024	0.0306	0.2301	0.0745	0.0473	0.3883
WorldVista	0.0784	0.049	0.3846	0.1055	0.0692	0.3961	0.0984	0.043	0.3041	0.0747	0.0418	0.3588
ZEPRS	0.0955	0.0224	0.19	0.14	0.0129	0.0844	0.106	0.0197	0.1567	0.088	0.0206	0.1897
ClearHealth	0.0837	0.0496	0.3721	0.127	0.0433	0.2543	0.0881	0.0673	0.4331	0.0759	0.0468	0.3814
MedinTux	0.0651	0.0695	0.5163	0.1018	0.0779	0.4335	0.0709	0.0812	0.5339	0.0623	0.0615	0.4968

Based on integrated AHP–SAW

Software	Scores with different developer weighted			
	W1	W2	W3	W4
FreeMED	0.528	0.563	0.407	0.463
GNUmed	0.739	0.634	0.835	0.847
GNU Health	0.671	0.546	0.637	0.713
Hospital OS	0.411	0.321	0.52	0.452
HOSxP	0.382	0.364	0.611	0.435
OpenEMR	0.849	0.883	0.826	0.901
OpenMRS	0.724	0.602	0.780	0.814
OSCAR	0.638	0.623	0.560	0.563
THIRRA	0.503	0.392	0.360	0.503
WorldVista	0.497	0.515	0.409	0.457
ZEPRS	0.289	0.242	0.260	0.278
ClearHealth	0.532	0.433	0.514	0.475
MedinTux	0.414	0.401	0.517	0.443

Based on integrated AHP–HAW

Software	Scores with different developer weighted			
	W1	W2	W3	W4
FreeMED	0.070	0.091	0.055	0.062
GNUmed	0.109	0.093	0.133	0.126
GNU Health	0.084	0.066	0.077	0.093
Hospital OS	0.045	0.034	0.060	0.054
HOSxP	0.060	0.061	0.109	0.065
OpenEMR	0.122	0.145	0.113	0.128
OpenMRS	0.102	0.086	0.103	0.115
OSCAR	0.089	0.091	0.076	0.078
THIRRA	0.059	0.045	0.039	0.064
WorldVista	0.060	0.065	0.049	0.055
ZEPRS	0.030	0.027	0.026	0.030
ClearHealth	0.069	0.056	0.070	0.063
MedinTux	0.120	0.128	0.074	0.080

Based on integrated AHP–WPM

Software	Scores with different developer weighted			
	W1	W2	W3	W4
FreeMED	7	7	7	7
GNUmed	11	11	11	11
GNU Health	8	8	8	8
Hospital OS	1	1	1	1
HOSxP	6	6	6	6
OpenEMR	12	12	12	12
OpenMRS	10	10	10	10
OSCAR	9	9	9	9
THIRRA	5	5	5	5
WorldVista	4	4	4	4
ZEPRS	2	2	2	2
ClearHealth	3	3	3	3
MedinTux	0	0	0	0

Table 4 (*continued*)

Based on integrated AHP–WSM				
Software	Scores with different developer weighted			
	W1	W2	W3	W4
FreeMED	1.533	1.119	1.347	1.211
GNUmed	2.012	1.491	2.194	1.886
GNU Health	2.009	1.377	2.004	1.749
Hospital OS	1.788	1.247	2.033	1.496
HOSxP	0.928	0.734	1.243	0.857
OpenEMR	2.331	1.822	2.386	2.042
OpenMRS	2.148	1.577	2.334	1.936
OSCAR	1.718	1.314	1.685	1.419
THIRRA	1.711	1.134	1.538	1.444
WorldVista	1.724	1.326	1.728	1.422
ZEPRS	1.232	0.824	1.192	1.002
ClearHealth	1.920	1.388	2.031	1.561
MedinTux	1.321	1.039	1.718	1.164

categories identified in the ISO 27799 standard [48]. Secure storage and backup mechanisms are well-established measures in research and industry literature, e.g. [49–52], respectively. In Table 2, a *Y* symbol was placed in the corresponding column along the systems' row if a system declared its compliance with HIPAA regulations. Similarly, if the system provided a security measure in any perceivable way, then the corresponding column would contain a check mark.

3.1.1.2.4. User support. User support can take many forms. Different users prefer different sources to learn about a system and obtain help. Some people would be comfortable referring to textual materials, whereas others are visual or auditory learners. Impatient or busy users would appreciate shorter tutorials and to-the-point instructions more than lengthy documents. For a product that extends to more than one discipline, such as a health information system where users might lack adequate information technology literacy, a committed technical support, even through paid options, is necessary. Therefore, documentation of a product usually and aptly varies in media, volume, and level of details. The support notion was broken into several components to provide a better view, and the studied systems were compared based on those elements; the results are presented in Table 2. Our set of user support sources is based on a comprehensive survey of any available online help for all the studied open-source software. From a user perspective, all these elements can provide useful information about the process of selecting, installing, and then using the software in production. A common practice among software providers for Web-based applications is to offer a demo implementation of the application that is hosted on the provider's own server or a donated hosting server. Some factors are well-known means of communicating with users (e.g., forums and Web log), whereas others are becoming more common among software developers (e.g., webcasts [41].)

3.1.1.2.5. Developer support. Similar to user support, support for developers can be regarded as a combination of several characteristics [46]. Table 2 enumerates the software systems and indicates their provision of each support feature. Availability of a discussion group to share ideas, questions, and answers is a common practice in open-source projects and provides a first resort when encountering undocumented issues during the development process. The same is true for mailing lists where subscribed developers can receive up-to-date news and view recent discussion threads. For new developers, a developer guide is essential to explain core modules, coding styles, and similar aspects specific to the target project. Comment density is also an established quality indicator for OSS [33]. The analysis of an open-source directory (www.ohloh.net) was the basis when the adequacy of code comments was estimated, where more than 10% of code lines were deemed an adequate comment percentage.

3.1.1.2.6. Customization. Considering that the systems under study were open-source software, modifying the software to the user's own needs is possible provided the required technical knowledge and documentation are available and affordable. However, the ability to customize the system from a user's perspective was considered. To help manage the comparison, the customizability was divided into aspects of localization, controlling the interface elements, add-on functionality, dynamic reporting, and the ability to run custom scripts. All measures were derived by inspecting the OSS applications under study. System-level customization that needs to be conducted by highly qualified personnel in a production environment (e.g. as pertaining to the operating system or Web server configurations) was not considered. Table 2 lists the details of the adopted components.

3.1.1.2.7. Ease of deployment. Deployment of software systems is a general term that comprises many activities and extends over a long period, from software installation to decommission. Attention is restricted within the most obvious and the conceivably most difficult aspects of installing the software, from a requirement-compliant computer system to a launched copy of the software on that system. Ease of installation is one of the measures included in the ISO 9126 standard [38]. The examined systems vary widely in the installation process. They range from being straightforward (i.e., single-button installation) to being frustrating (i.e., spending several days trying to figure out what went wrong.) Most of the systems assume a minimum level of technical literacy; they support those who are at least familiar with setting up common software libraries and database engines. Part of the difficulty we encountered may be due to the variety of the systems' technologies, which calls for broader expertise with different settings, rather than a difficulty in the systems themselves. A measure that can assess the

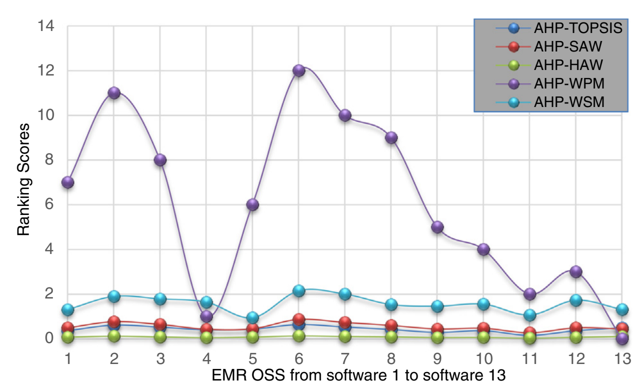

Fig. 3. Ranking scores based on different integrated AHP–(WPM/WSM/SAW/HAW/TOPSIS).

Table 5
P value illustrated from paired sample t-test on ranking scores.

AHP–WPM	AHP–HAW	AHP–SAW	AHP–TOPSIS	
			8.75858E−05	AHP–SAW
		2.6848E−08	2.53323E−08	AHP–HAW
	6.69339E−05	0.000102896	9.87191E−05	AHP–WPM
0.000493	9.87059E−10	1.03534E−09	2.57403E−09	AHP–WSM

relative ease of installing each examined system is identified to facilitate the comparison of the systems' installations. The ease with which software can be installed is interpreted as the time required for installation. This element is derived from our own experience installing the sample OSS under analysis. The results of the system comparison are summarized based on the measures in Table 2.

As mentioned in the result evaluation, the following conclusions are confirmed:

- Most of the systems are similar on the usability scale, except for those systems with non-English interface. Users with more IT experience tend to give lower satisfaction rate for the subject EMR applications, whereas a specific user with pure medical background is more impressed by the electronic systems and feeds higher satisfaction scores, although this user spends relatively more time accomplishing the tasks.
- The most common security measure in our set of systems is authentication and access control [28], which mainly utilizes passwords. Secure storage and secure communications are totally absent from current open-source systems, whereas some systems have already begun to comply with HIPAA privacy rules. Unsurprisingly, a few systems even lack security or privacy mechanism, because applications of emerging fields tend to first focus on functionality or usability and then consider security.
- With regard to user support criteria, almost all systems have a website; the worst systems in this respect are those directed toward specific audiences and not the general English-speaking public. Notably, a few systems (e.g. OpenEMR) satisfy all our measures of user support [25,26,29]. Availability of a demo website is a common practice among Web-based systems, and most Web-based open-source systems provide this option.
- Support for developers is less than the support for users. Nevertheless, most systems, except those of an individual, governmental, or project-oriented nature, have code repositories, as expected from open-source projects. Similar to the security measure, much support is needed for developers in available open-source healthcare applications.
- GNUmed by far is the most customizable system, whereas GNU and popular open-source projects (community-wise) similar to OpenEMR provide support localization options. Many systems enable the user to change some aspects of appearance, including color theme and fonts; add custom plug-ins that are possibly developed by third parties; and extend the functionality of the open-source application.
- The reported installation time measures only the consumed time during the actual installation, excluding the time spent in frustration because of troubleshooting or learning about the process. The installation time is relatively low, but the level of expected expertise is somewhat high in at least half of the systems. Although instructions are usually provided, a certain level of familiarity with setting Web and database servers is implied. Time ranges (in hours) are 1–[0–1/4], 2–[1/4–1/2], 3–[1/2–3/4], 4–[3/4–1], 5–[1–1 1/4], 6–[1 1/4–1 1/2], and 7–[1 1/2–1 3/4]. Time might be taken with tolerance, as many operations depend on network speed.

- *Selecting open-source EMR software packages*
The OS-EMR software was evaluated using several criteria, including functionality, usability, security, user support, developer support, and ease of installation. Another problem emerged, that is, each software has several attributes, and each decision maker has different weights for these attributes. Therefore, selecting which software is suitable per use is difficult. On the one hand, users who aim to use one kind of software might give more weight to functionality, usability, and user support rather than other features that gain less interest than these attributes, whereas users who aim to develop this software in real clinics would probably target different attributes. On the other hand, EMR software package selection (in particular open-source software) is a multi-attribute problem where each kind of software is considered an available alternative for the decision maker. To solve this problem, developing AHP multi-attribute decision making to calculate the weight of each attribute and using WPM/WSM/SAW/HAW/TOPSIS to rank the available alternatives can be considered. Based on different integrations between MCDM techniques, AHP is a widely used algorithm to solve complex multi-attribute selection problems in different life areas. The selection part can be described as follows:

Step 1: Weights are assigned to each basic attribute in the hierarchy of criteria using AHP. Each basic criterion is rated in the hierarchy for each OS-EMR software considered for evaluation. AHP is used to derive ratio scales from pairwise comparisons. Four participants who are software developers with more than three years of experience are selected to complete the AHP. Four copies of pairwise comparisons with a total of 98

Table 6
Representing decision-making techniques scores to the ranking orders.

Software	Technique ranking scores based on AHP and					Technique ranking order based on AHP and				
	TOPSIS	SAW	HAW	WPM	WSM	TOPSIS	SAW	HAW	WPM	WSM
FreeMED	0.3609	0.49025	0.0695	7	1.3025	4	8	6	8	3
GNUmed	0.608975	0.76375	0.11525	11	1.89575	12	12	12	12	11
GNU Health	0.52065	0.64175	0.08	8	1.78475	10	10	8	9	10
Hospital OS	0.402025	0.426	0.04825	1	1.641	6	2	2	2	8
HOSxP	0.42925	0.448	0.07375	6	0.9405	8	5	7	7	1
OpenEMR	0.63595	0.86475	0.127	12	2.14525	13	13	13	13	13
OpenMRS	0.525175	0.73	0.1015	10	1.99875	11	11	11	11	12
OSCAR	0.416025	0.596	0.0835	9	1.534	7	9	9	10	6
THIRRA	0.2841	0.4395	0.05175	5	1.45675	2	3	3	6	5
WorldVista	0.3609	0.4695	0.05725	4	1.55	4	6	4	5	7
ZEPRS	0.1552	0.26725	0.02825	2	1.0625	1	1	1	3	2
ClearHealth	0.360225	0.4885	0.0645	3	1.725	3	7	5	4	9
MedinTux	0.495125	0.44375	0.1005	0	1.3105	9	4	10	1	4

Table 7
P value illustrated from paired sample t-test on ranking orders.

AHP–WPM	AHP–HAW	AHP–SAW	AHP–TOPSIS	
			0.460703	AHP–SAW
		0.5	0.436778	AHP–HAW
	0.5	0.5	0.466981	AHP–WPM
0.5	0.5	0.5	0.467958	AHP–WSM

comparisons between all criteria and sub-criteria are presented to software developers, and their responses on these criteria are obtained. The relative scale (1 to 9) is created to measure how much the developer likes a criterion compared with other criteria. Each developer critically analyzes these criteria according to their experience and knowledge. After attaining the responses on pairwise comparisons, the reciprocal matrix is created from the pairwise comparisons. Finally, the eigenvector is computed, which provides the relative ranking of the criteria as shown in Table 3.

Step 2: The available alternative scores are ranked in descending order, and the best software is selected based on WPM/WSM/SAW/HAW/TOPSIS. Aggregate scores only provide an idea about which software package is better than the other. Individuals can always be relied upon to select the best software package, as in other selection options.

4. Results and discussion

The values for evaluation metric are presented as follows: security; user support; developer support; and customizability, which present the assigned true value as a scale of 1 and the assigned untrue value as a scale of 0. Usability and ease of installation are used as the same values (Table 2). The experimental results on the functionality and features are classified into four domains, namely, patient handling, medical records, support data, and admin tools. Each of these domains includes sub-features numbered from 1 to 25. The total scores of each application are combined for these features to be distributed in the OS-EMR software systems as OpenEMR (19/25), GNUmed (14/25), GNUHealth (16/25), WorldVista (14/25), OSCAR (14/25), FreeMED (13/25), HOSxP (10/25), HospitalOS (9/25), MedinTux (9/25), OpenMRS (8/25), ClearHealth (7/25), THIRRA (7/25), and ZEPRS (5/25). Five groups of experiments based on the evaluation metric are executed differently for each of the following techniques: AHP–SAW, AHP–HAW, AHP–WPM, AHP–WSM, and AHP–TOPSIS. For all presented techniques, the scores that applied the weight of perspective from Developers 1 to 4,

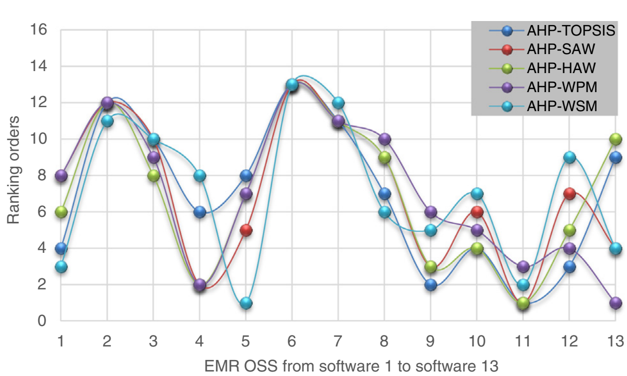

Fig. 4. Ranking orders based on different integrated AHP–(WPM/WSM/SAW/HAW/TOPSIS).

represented as W1 to W4, respectively, are shown under "Scores with different developer weighted" (Table 4).

For each technique, using the values given in Table 4, the average of the four ranking scores weighted from different perspectives of the developers is presented in Fig. 3. This chart shows the variation in the ranking scores of all OS-EMR software packages with the weight factor. The x-axis represents OS-EMR software, FreeMED, GNUmed, GNUHealth, Hospital OS, HOSxP, OpenEMR, OpenMRS, OSCAR, THIRRA, WorldVista, ZEPRS, ClearHealth, and MedinTux from Software 1 to 13. The y-axis in Fig. 3 represents the results from different integrations of AHP (WPM/WSM/SAW/HAW/TOPSIS).

Using different decision-making techniques result in different ranking scores per alternative, as shown in Fig. 3. This result is due to the differences in the mathematical processes for each technique. Paired sample t-test is utilized to compare different techniques. This test is a statistical technique that is used to compare two samples to measure correlations. The results show significant differences among all decision-making techniques, in which all p values < 0.05 (see Table 5). These results are attributed to the ranking scores from different techniques with different scales.

Unifying the scales of decision-making techniques can give valuable comparison among them. This action can be conducted by representing the ranking scores to the ranking orders in place of the technique scale, as shown in Table 6. We convert the score results measured by each technique into ranking order data. The highest order (i.e., 13) is given to the highest score and so on.

Table 6 shows that the ranking orders between AHP–TOPSIS and AHP–SAW are equal in 5 software and different in 8 software. Similar result is obtained for AHP–TOPSIS and AHP–HAW. The ranking orders between AHP–TOPSIS and AHP–WPM are equal in 3 software and different in 10 software. The ranking orders between AHP–TOPSIS and AHP–WSM are equal in 2 software and different in 11 software. Comparing AHP–SAW with AHP–HAW, we observe that 6 alternatives carry the same ranking orders and 7 software carry different ranking orders. AHP–SAW has 4 alternatives carrying the same ranking orders with AHP–WPM, whereas AHP–WSM has 2 alternatives carrying the same ranking orders with AHP–SAW. Comparison of AHP–HAW and AHP–WSM shows that 2 alternatives carry the score orders. Finally, we observed that only 1 alternative carries the same scoring order between AHP–WPM and AHP–WSM.

Paired sample t-test is again utilized to measure the correlations among different techniques on ranking orders. The results show no significant differences among all decision-making techniques, in which all p values > 0.05 (see Table 7).

Figs. 3 and 4 indicate that the system numbers 2, 6, 7, and 11 do not differ in ranking scores/orders of experiments for all MCDM techniques presented. However, discrepancies among the ranking scores/orders are more noticeable in other systems, particularly 1, 3, 4, 5, 8, 9, 10, 12, and 13. From this summary of figures, we may conclude that several systems potentially serve as a good basis for ranking scores/orders. System 6, in particular, is an outstanding application. However, this system number is inadequate in providing customization and support for developers. Similarly, system 2 is a promising product because of its security aspects. System 7 has been deployed in many developing countries and is a capable application when the full spectrum of functionalities is covered beyond its original scenarios in resource-constrained environments. However, system 11 is not recommended because it is bound to the initial project for which it was developed. This system may be less flexible in adapting to a new situation. This argument becomes pertinent because of the unique underlying technology in terms of the database system and programming language.

5. Limitations of the research

This study has a thorough methodology and represents a relevant contribution to help decision-making processes in software application

selection. Nevertheless, two limitations are found on the samples of open-source software that form the target of this study. First, the list of included software is not comprehensive. However, such a list is representative of current active and popular projects as of the time of study. The exact list was compiled from a more compact inventory in [17] (under the electronic health or medical record category) to support a manageable and valid software sample. Second, this list was selected in August 2013. In the open-source world, considerable change could be expected in the span of one and a half year, including the rise and fall of projects. As of March 2015, the study sample still forms almost the same list in [17]. Further studies are required to keep the evaluation up to date because many OS-EMR systems might be updated and/or added within the coming years. The highlight of this study is not the results per se as much as the description of the application of a sound evaluation methodology to a specific type of software application. This study should serve as an example for using the evaluation methodology for the OS-EMR software.

6. Research contributions

A comparative study for the OS-EMR software is conducted, which provides the following results:

- Significant differences exist among MCDM techniques based on different integrations on ranking scores, in which all p values < 0.05. By contrast, no significant differences exist among them when the ranking scores are represented to the ranking orders in place of the technique scale, in which all p values > 0.05.
- The software GNUmed, OpenEMR, OpenMRS, and ZEPRS do not differ in ranking scores/orders of experiments for all MCDM techniques presented. However, discrepancies among the ranking scores/orders are more noticeable in other software.
- GNUmed, OpenEMR, and OpenMRS software are the most promising candidates for providing a good basis on ranking scores/orders, whereas ZEPRS is not recommended because it records the worst ranking score/order in comparison with other OS-EMR software.

7. Conclusion

A hands-on study was performed in this research. A set of applications was locally implemented on separate virtual machines to examine OS-EMR software alternatives closely. Several measures were set as bases for comparison, and the systems were scored on the basis of a set of metric outcomes using different integrations of AHP (WPM/WSM/SAW/HAW/TOPSIS) in group contents from the available options of the OS-EMR software. Paired sample t-test was applied to measure the correlations among different techniques on ranking scores/orders. Findings show significant differences among MCDM techniques based on different integrations on ranking scores, in which all p values < 0.05. On the contrary, no significant differences exist among them when the ranking scores are represented to the ranking orders in place of the technique scale, in which all p values > 0.05. The software GNUmed, OpenEMR, OpenMRS, and ZEPRS do not differ in ranking scores/orders of experiments for all MCDM techniques presented. However, discrepancies among the ranking scores/orders are more noticeable in other software. GNUmed, OpenEMR, and OpenMRS software are the most promising candidates for providing a good basis on ranking scores/orders, whereas ZEPRS is not recommended because it records the worst ranking score/order in comparison with other OS-EMR software. Further investigation is required, particularly on security, interoperability, and developer support. The systems, as software applications, are similar in terms of usability aspect and share a common set of functionality, although they vary considerably in community support and activity.

Acknowledgments

This research was funded by the High Impact Research Unit of the University of Malaya, under grant number UM.C/HIR/MOHE/FCSIT/12. We would like to express our very great appreciation to Assoc. Prof. Dr. Paul Hansen from University of Otago, New Zealand for his valuable comments and constructive suggestions during the planning and development of this research. His generosity with his time is very much appreciated.

References

[1] J. Verville, A. Hallingten, An investigation of the decision process for selecting an ERP software: the case of ESC, Management Decision 40 (3) (2002) 206–216.
[2] B. Hecht, Choose the right ERP software, Datamation 43 (3) (1997) 56–58.
[3] G. Salvatore, Multiple criteria decision analysis: state of the art surveys, International Series in Operations Research & Management Science 78 (2005) 1048.
[4] J. Figueira, S. Greco, M. Ehrgott, Multiple Criteria Decision Analysis: State of the Art Surveys, vol. 78, Springer, 2005.
[5] X.-F. Wang, J.-Q. Wang, S.-Y. Deng, A method to dynamic stochastic multicriteria decision making with log-normally distributed random variables, The Scientific World Journal 2013 (2013) (Article ID 202085, 8 pages).
[6] N. Al-Safwani, S. Hassan, N. Katuk, A multiple attribute decision making for improving information security control assessment, International Journal of Computer Applications 89 (3) (2014) 19–24.
[7] E. Triantaphyllou, B. Shu, S. Nieto Sanchez, T. Ray, Multi-criteria decision making: an operations research approach, in: J.G. Webster (Ed.), Encyclopedia of Electrical and Electronics Engineering, vol. 15, John Wiley & Sons, New York, NY 1998, pp. 175–186.
[8] E. Triantaphyllou, Multi-criteria Decision Making Methods: A Comparative Study, 1st edition Springer Science, Kluwer Academic Publishers, 2000. 290.
[9] M. Aruldoss, T.M. Lakshmi, V.P. Venkatesan, A survey on multi criteria decision making methods and its applications, American Journal of Information Systems 1 (1) (2013) 31–43.
[10] K. Yoon, C. Hwang, Multiple Attribute Decision-making: An Introduction, Sage Publisher, 1995.
[11] J. Adler-Milstein, C.E. Green, D.W. Bates, A survey analysis suggests that electronic health records will yield revenue gains for some practices and losses for many, Health Affairs 32 (2013) 562–570.
[12] G.S. Kantor, W.D. Wilson, A. Midgley, Open-source software and the primary care EMR, Journal of the American Medical Informatics Association 10 (2003) 616-616.
[13] P.C. Webster, The rise of open-source electronic health records, The Lancet 377 (2011) 1641–1642.
[14] M.J. Doyle, Open source will help drive EHR costs down. The use of open source in healthcare will break down many barriers, from high cost and lack of interoperability, to inaccessibility and complexity, Health Management Technology 30 (2009) 10–11.
[15] W. Ross, T. Jones, L. Trigg, S. Renly, M. Hogarth, N.H. Arzt, Free and Open Source, Software in Healthcare 1.0, 2008.
[16] C.J. Reynolds, J.C. Wyatt, Open source, open standards, and health care information systems, Journal of Medical Internet Research 13 (2011).
[17] List of open-source healthcare softwareRetrieved 11/08/2013, from http://www.ourmed.org/wiki/List_of_open_source_healthcare_software.
[18] Medical Free/Libre and Open source SoftwareRetrieved 11/08/2013, from http://www.medfloss.org/node/614.
[19] A.A. Zaidan, B.B. Zaidan, Ahmed Al-Haiqi, M.L.M. Kiah, Hussaen Muzamel, Mohamed Abdulnabi, Evaluation and selection of open-source EMR software packages based on integrated AHP and TOPSIS, Journal of Biomedical Informatics 53 (2015) 390–404.
[20] M. Mollaghasemi, J. Pet-Edwards, Technical Briefing: Making Multiple Objective Decisions, IEEE Computer Society Press, Los Alamitos, California, 1997.
[21] T.L. Saaty, M.S. Ozdemir, Why the magic number seven plus or minus two, Mathematical and Computer Modelling 38 (2003) 233–244.
[22] A.S. Jadhav, R.M. Sonar, Evaluating and selecting software packages: a review, Information and Software Technology 51 (2009) 555–563.
[23] C. Ncube, J.C. Dean, The limitations of current decision making techniques in the procurement of COTS software components, Proceedings of the First International Conference on COTS-based Software System, Orlando, February 2002 2002, pp. 176–187.
[24] A.S. Jadhav, R.M. Sonar, Framework for evaluation and selection of the software packages: a hybrid knowledge based system approach, Journal of Systems and Software 84 (8) (2011) 1394–1407, http://dx.doi.org/10.1016/j.jss.2011.03.034.
[25] N.A. Kalogriopoulos, J. Baran, A.J. Nimunkar, J.G. Webster, Electronic medical record systems for developing countries: review, Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE 2009, pp. 1730–1733.
[26] C.J. Seebregts, B.W. Mamlin, P.G. Biondich, H.S. Fraser, B.A. Wolfe, D. Jazayeri, C. Allen, J. Miranda, E. Baker, N. Musinguzi, et al., The OpenMRS implementers network, International Journal of Medical Informatics 78 (2009) 711–720.
[27] A. Seffah, M. Donyaee, R.B. Kline, H.K. Padda, Usability measurement and metrics: a consolidated model, Software Quality Journal 14 (2006) 159–178.

[28] E. Helms, L. Williams, Evaluating access control of open source electronic health record systems, Proceedings of the 3rd Workshop on Software Engineering in Health Care 2011, pp. 63–70.
[29] N.A. Mohammed-Rajput, D.C. Smith, B. Mamlin, P. Biondich, B.N. Doebbeling, et al., OpenMRS, a global medical records system collaborative: factors influencing successful implementation, AMIA Annual Symposium Proceedings, vol. 2011 2011, p. 960.
[30] A. Jadhav, R. Sonar, Analytic hierarchy process (AHP), weighted scoring method (WSM), and hybrid knowledge based system (HKBS) for software selection: a comparative study, 2009 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET) 2009, pp. 991–997.
[31] T.L. Saaty, Decision making with the analytic hierarchy process, International Journal of Services Sciences 1 (1) (2008) 83–98.
[32] M.L.M. Kiah, A. Haiqi, B.B. Zaidan, A.A. Zaidan, Open source EMR software: profiling, insights and hands-on analysis, Computer Methods and Programs in Biomedicine 117 (2) (2014) 360–382.
[33] O. Arafat, D. Riehle, The comment density of open source software code, 31st International Conference on Software Engineering-Companion Volume, 2009. ICSE-Companion 2009 2009, pp. 195–198.
[34] B. Janamanchi, E. Katsamakas, W. Raghupathi, W. Gao, The state and profile of open source software projects in health and medical informatics, International Journal of Medical Informatics 78 (2009) 457–472.
[35] S. Kobayashi, Open source software development on medical domain, in: Christos Kalloniatis (Ed.), Modern Information Systems 2012, pp. 953–978.
[36] I. Maglogiannis, Towards the adoption of open source and open access electronic health record systems, Journal of Healthcare Engineering 3 (2012) 141–162.
[37] F. Zuniga, A. Enrique, K. Win, W. Susilo, Functionalities of free and open electronic health record systems, International Journal of Technology Assessment in Health Care 26 (2010) 382–389.
[38] I. O. for Standardization/International Electrotechnical Commission, et al., ISO/IEC 9126-1 Standard, Software Engineering, Product Quality, Part 1: Quality Model, 2001.
[39] X. Franch, J.P. Carvallo, Using quality models in software package selection, Software, IEEE, vol. 202003. 34–41.
[40] D.A. Wheeler, How to evaluate open source software/free software (OSS/FS) programsURL http://www. dwheeler. com/oss_fs_eval. html2007.
[41] C. Dufour, E.G. Toms, J. Lewis, R. Baecker, User strategies for handling information tasks in webcasts, CHI'05 Extended Abstracts on Human Factors in Computing Systems 2005, pp. 1343–1346.
[42] Addison-Wesley Professional; 1 edition (April 17, 1999), 608 pages.
[43] J. Nielsen, Usability Engineering, vol. 1, AP Professional, Boston, 1993. (c1993).
[44] B. Shackel, Usability-context, framework, definition, design and evaluation, Human Factors for Informatics Usability 1991, pp. 21–37.
[45] L.L. Constantine, L.A. Lockwood, Software for Use: A Practical Guide to the Models and Methods of Usage-centered Design, Pearson Education, 1999.
[46] G. von Krogh, S. Spaeth, The open source software phenomenon: characteristics that promote research, The Journal of Strategic Information Systems 16 (2007) 236–253.
[47] H.O. Alanizi, M.L. Mat Kiah, A.A. Zaidan, B.B. Zaidan, G.H. Alam, Secure topology for electronic medical record transmissions, International Journal of Pharmacology 6 (2010) 954–958.
[48] E. ISO, 27799: 2008 Health Informatics, Information Security Management in Health Using ISO/IEC, vol. 270022008.
[49] M.S. Nabi, M.L. Mat Kiah, B.B. Zaidan, A.A. Zaidan, G.M. Alam, Suitability of using SOAP protocol to secure electronic medical record databases transmission, International Journal of Pharmacology 6 (2010) 959–964.
[50] R. Mogull, Security Requirements for Electronic Medical Records, 2009.
[51] M.L. Kiah, M.S. Nabi, B.B. Zaidan, A.A. Zaidan, An enhanced security solution for electronic medical records based on AES hybrid technique with SOAP/XML and SHA-1, Journal of Medical Systems 37 (2013) 1–18.
[52] H.O. Alanazi, A.A. Zaidan, B.B. Zaidan, M.L. Kiah, S.H. Al-Bakri, Meeting the security requirements of electronic medical records in the ERA of high-speed computing, Journal of Medical Systems 39 (2015) 1–13.

A.A. Zaidan received his first class B. Eng. degree in Computer Engineering in 2004 from University of Technology, Baghdad, Iraq. Then, he received his M.Sc. degree on Data Communications and computer network in 2009 from University of Malaya, Malaysia. Then, following his Ph.D. he received the degree on artificial intelligence in 2013 from Multimedia University, Malaysia. Currently, he is working as senior lecturer at Department of Computing, University Pendidikan Sultan Idris. He led or is a member of many funded research projects and he has published more than 110 papers at various international conferences and journals. His research areas are: Machine Learning and Multiple-criteria Decision-making.

B.B. Zaidan received his B.Sc. in Applied Mathematics in 2004 from Saddam University (currently Al-Nahrain University), Baghdad, Iraq. In 2009. He received his M.Sc. in Data Communications and Information Security from University of Malaya, Malaysia. Recently, he is completing his Ph.D. at Multimedia University, Malaysia. He has published more than 110 papers at various international conferences and journals. His research interests include Risk Analysis, Security Policy, AI Applications on Security, Cyber Security, Telemedicine Security, and Security Analysis of e-health and m-health.

Muzammil Hussain received his B.S. (Computer Science) in 2013 from COMSATS Institute of Information Technology Sahiwal, Pakistan. Currently, he is pursuing his Ph.D. in Computer Science at University of Malaya, Kuala Lumpur, Malaysia. He has published 3 international journal papers. His current research interests include Android Security, Operating System Security, mHealth Security, Mobile Security, Sensors Threats, and Telemedicine Security.

Ahmed Mubarak Al-Haiqi received his Ph.D. in Electrical Engineering from National University of Malaysia (UKM), Malaysia, where he has obtained the Master of Communications and Computer Engineering degree in 2010. His current research interests are the smartphone — in particular, android — security, sensor threats, and sensor programming. Previously he had some research in network security, and before many years, he used to be a database developer.

M.L. Mat Kiah received her B.Sc. (Hons) in Computer Science from University of Malaya (UM), Malaysia in 1997, M.Sc. in 1998 and Ph.D. in 2007 from Royal Holloway, University of London, UK. She joined the Faculty of Computer Science & Information Technology, UM as a tutor in 1997. Her current research interests include key management, secure group communication and wireless mobile security. She is also interested in routing protocols and mobile ad-hoc networks. A total of 42 (journal: 16, conference: 11, book chapter: 01) publications are attributed to her name.

Mohamed Abdulnabi is currently a Ph.D. candidate and a research assistant in secure framework for healthcare networks, a research project funded by the Malaysian Ministry of Higher Education. He received his B.Sc. (Computer Science) in 2006 from India and MCS (Computer Science) in 2012 from University of Malaya, Malaysia. He worked as lecturer at the faculty of computer science in University of Modern Sciences from 2007–2009 in Yemen. Mohamed has his started publication journey by publishing his first article in ISI Web of Science in 2010, later he has published a number of conferences and articles in most prestigious journals internationally. His research interests include information security, healthcare networks, hospital information systems and secure android medical apps.